

LAPORAN JOBSHEET 8

Dasar Pemrograman

Fungsi dan Rekursif

Dibina oleh Ibu Vivi Nur Wijyaningrum, S.Kom., M.Kom.



Disusun oleh :

Fitria Ramadhani Prihandiva

D-IV Sistem Informasi Bisnis 1C / 14

2241760055

POLITEKNIK NEGERI MALANG

JURUSAN TEKNOLOGI INFORMASI

TAHUN 2023

Nama : Fitria Ramadhani Prihandiva
Kelas : SIB 1C
NIM : 2241760055
Mata Kuliah : Praktikum Dasar Pemrograman

JOBSHEET 9

Percobaan 1 : Membuat Fungsi Tanpa Parameter

```
D: > College > SEMESTER 2 > Daspro > Jobsheet 9 > J Greeting14.java > ...  
1 public class Greeting14{  
2     static void berisalam(){  
3         System.out.println("Halo! Selamat Pagi");  
4     }  
5     public static void main (String[]args){  
6         berisalam();  
7     }  
8 }  
9
```

```
PS C:\Users\FITRIA RAMADHANI> & 'C:\Program Files\Java  
\jdk-19\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionM  
essages' '-cp' 'C:\Users\FITRIA RAMADHANI\AppData\Local  
\Temp\vscodesws_d20e0\jdt_ws\jdt.ls-java-project\bin' '  
Greeting14'  
Halo! Selamat Pagi  
PS C:\Users\FITRIA RAMADHANI>
```

Pertanyaan!

1. Apakah fungsi tanpa parameter selalu digunakan untuk fungsi “void” saja?

Jawaban : Tidak, fungsi tanpa parameter tidak selalu digunakan hanya untuk fungsi "void". Fungsi tanpa parameter dapat digunakan untuk fungsi dengan tipe pengembalian apa pun, termasuk tipe data selain void. **Fungsi tanpa parameter digunakan ketika tidak ada argumen/data yang perlu diteruskan ke dalam fungsi tersebut.** Ini dapat berlaku untuk berbagai jenis fungsi, tidak hanya fungsi "void".

2. Apakah bisa kalimat “Halo! Selamat Pagi” dituliskan tanpa fungsi berisalam? Modifikasi kode program tersebut untuk dapat menampilkan suatu kalimat **tanpa dan dengan** menggunakan fungsi!

```
D: > College > SEMESTER 2 > Daspro > Jobsheet 9 > J Greeting14.java > Greeting14 > main(String[])  
1 public class Greeting14{  
2     static void berisalam(){  
3         System.out.println("Halo! Selamat Pagi");  
4     }  
5     public static void main (String[]args){  
6         berisalam();  
7         String Modifikasi = "Halo! Selamat Pagi (Modifikasi)";  
8         System.out.println (Modifikasi);  
9     }  
10 }  
11
```

Halo! Selamat Pagi
Halo! Selamat Pagi (Modifikasi)

3. Apakah keuntungan menggunakan fungsi di dalam program?

- Karena setiap mempunyai fokus kerja masing masing sehingga lebih mudah dalam mengerjakan
- Mempermudah dalam pemanggilan fungsi yang diinginkan
- Membaca kode program lebih mudah dan dipahami

Percobaan 2 : Membuat Fungsi Dengan Parameter

```
1 public class Greeting14{
2     static void beriSalam(){
3         System.out.println("Halo! Selamat Pagi");
4     }
5     static void beriUcapan(String ucapan){
6         System.out.println (ucapan);
7     }
8     public static void main (String[] args){
9         beriSalam();
10        String Modifikasi = "Halo! Selamat Pagi (Modifikasi)";
11        System.out.println (Modifikasi);
12        String salam = "Selamat datang di pemrograman Java";
13        beriUcapan (salam);
14    }
15 }
```

```
Halo! Selamat Pagi
Halo! Selamat Pagi (Modifikasi)
Selamat datang di pemrograman Java
```

Pertanyaan!

1. Apakah kegunaan suatu parameter di dalam fungsi?

Jawaban :

- Menerima input data dari luar fungsi
- Mengirim data antar fungsi satu ke fungsi lainnya
- Penentuan tipe data yang masuk sebagai input an

2. Apakah parameter sama dengan variabel? jelaskan!

Jawaban : Menurut saya berbeda. Parameter bertujuan untuk mengenalkan pada fungsi data apa yang dari luar fungsi tersebut kemudian akan masuk kedalam fungsi tersebut. Sedangkan Variable lebih ke entitas yang bisa menyimpan data datanya sesuai dengan tipenya.

3. Apakah parameter di dalam Bahasa pemrograman Java hanya digunakan untuk melewati data input saja? Bagaimana dengan data output?

Jawaban : Tidak, parameter bisa digunakan untuk mengirim data output ke pemanggil fungsi (main)

Percobaan 3 : Membuat Fungsi Dengan Nilai Kembalian

```
1 public class Persegi14 {
2     static int luasPersegi(int sisi){
3         int luas = sisi*sisi;
4         return luas;
5     }
6     public static void main (String[] args){
7         int luasan = luasPersegi(sisi:5);
8         System.out.println("Luas Persegi dengan sisi 5 = " + luasan);
9     }
10 }
```

```
Luas Persegi dengan sisi 5 = 25
PS C:\Users\FITRIA RAMADHANI>
```

Pertanyaan!

1. Jelaskan kapan suatu fungsi membutuhkan nilai kembalian (return)!

Jawaban : Jika suatu fungsi tersebut memiliki nilai hasil suatu operasi seperti hitungan matematika, hasil tersebut merupakan data yang harus tersimpan di suatu variable dan ditangkap oleh fungsi main sehingga bisa terpanggil.

2. Apakah bisa ditambahkan *statement* **System.out.println** di dalam fungsi dengan nilai kembalian? Apa pengaruhnya?

```
public class Persegi14 {
    static int luasPersegi(int sisi){
        int luas = sisi*sisi;
        return luas;
        System.out.println("Test");
    }
}
```

Exception in thread "main" java.lang.Error: Unresolved compilation problem: Unreachable code

at Persegi14.luasPersegi(Persegi14.java:5)

at Persegi14.main(Persegi14.java:8)

```
blic class Persegi14 {
    static int luasPersegi(int sisi){
        int luas = sisi*sisi;
        System.out.println("Test");
        return luas;
    }
}
```

```
Test
Luas Persegi dengan sisi 5 = 25
PS C:\Users\FITRIA RAMADHANI>
```

Jawaban : Bisa selama posisi nya masih sebelum return. Sedangkan apabila diletakkan setelah return maka *statement* **System.out.println** tidak tereksekusi dan menyebabkan error.

3. Apakah bisa fungsi tanpa nilai kembalian di dalam fungsi main dipanggil tanpa dilewatkan ke variabel, seperti pada percobaan 1? Jelaskan!

Jawaban :Seperti pada percobaan 1, Bisa karena tidak punya kembalian dan langsung diterima perintah Print maka bisa langsung dijalankan.

Kalau mempunyai kembalian tidak bisa, karena pada proses pemanggilannya Sebelum dipanggil, Kembalian harus ditangkap dan disimpan di fungsi main (Fungsi main harus tahu hasilnya dulu) baru bisa dilakukan pemanggilan dan ditampilkan.

Percobaan 4 : Fungsi Varargs

```
1 public class Percobaan414 {
2     static void Tampil(String str,int... a){
3         System.out.println("String: " + str);
4         System.out.println("Jumlah argumen/parameter: " + a.length);
5
6         for (int i : a){
7             System.out.print(i + " ");
8         }
9         System.out.println();
10    }
11    public static void main (String[]args){
12        Tampil(str:"Daspro 2019", ...a:100, 200);
13        Tampil(str:"Teknologi Informasi", ...a:1,2,3,4,5);
14        Tampil(str:"Polinema");
15    }
16 }
```

```
String: Daspro 2019
Jumlah argumen/parameter: 2
100 200
String: Teknologi Informasi
Jumlah argumen/parameter: 5
1 2 3 4 5
String: Polinema
Jumlah argumen/parameter: 0
```

Pertanyaan!

1. Jelaskan mengapa penulisan parameter di praktikum 5 di tulis dengan **int... a**!

Jawaban : Karena tidak mengetahui panjang/berapa banyak data sehingga untuk mempersingkat penulisan juga menggunakan **int...a**

2. Sebutkan kegunaan varargs dalam implementasi kode program untuk menyelesaikan permasalahan dalam dunia nyata! (minimal 3)

- Varargs memungkinkan pengguna untuk memberikan jumlah argumen yang fleksibel saat memanggil fungsi. Ini berguna ketika Anda ingin mengizinkan pengguna untuk memberikan input dalam jumlah yang bervariasi.
- Varargs dapat digunakan untuk menghitung operasi matematika pada sejumlah angka yang bervariasi.
- Varargs dapat digunakan untuk menangani daftar data dengan jumlah elemen yang tidak diketahui sebelumnya

3. Bisakah kita menggunakan dua tipe data varargs dalam satu fungsi? Berikan contohnya!

Jawaban : Tidak bisa, satu fungsi hanya bisa menggunakan satu jenis Varargs, karena satu Varrgs harus memiliki tipe data yang sama. Kalau berbeda bisa gunakan Array dan ditempatkan di posisi sebagai Array.

Percobaan 5 : Fungsi Rekursif Menghitung Pangkat

```
1  import java.util.Scanner;
2  public class Percobaan514 {
3      static int hitungPangkat (int x, int y ){
4          if (y==0){
5              return (1);
6          }
7          else {
8              return(x*hitungPangkat(x, y-1));
9          }
10     }
11     public static void main(String[]args){
12         int hitungPangkat;
13         Scanner fit = new Scanner(System.in);
14         int bilangan,pangkat,hasil;
15         System.out.print(s:"Bilangan yang dihitung: ");
16         bilangan = fit.nextInt();
17         System.out.print(s:"Pangkat: ");
18         pangkat = fit.nextInt();
19         hasil = hitungPangkat(bilangan,pangkat);
20         System.out.print("Hasilnya adalah = " + hasil);
21     }
22 }
23
```

```
Bilangan yang dihitung: 3
Pangkat: 2
Hasilnya adalah = 9
```

Pertanyaan!

1. Terdapat pemanggilan fungsi rekursif **hitungPangkat(bilangan, pangkat)** pada fungsi main, kemudian dilakukan pemanggilan fungsi **hitungPangkat()** secara berulang kali. Jelaskan sampai kapan proses pemanggilan fungsi tersebut akan dijalankan!

Jawaban :

Sampai mendekati batas base case yang ditentukan maka perulangan akan berhenti dengan sendirinya lalu melakukan return untuk menghitung hasilnya

2. Tambahkan kode program untuk mencetak deret perhitungan pangkatnya. Contoh : **hitungPangkat(2,5)** dicetak $2 \times 2 \times 2 \times 2 \times 2 = 32$

```

public static void main(String[]args){
    int hitungPangkat;
    Scanner fit = new Scanner(System.in);
    int bilangan,pangkat,hasil;
    System.out.print(s:"Bilangan yang dihitung: ");
    bilangan = fit.nextInt();
    System.out.print(s:"Pangkat: ");
    pangkat = fit.nextInt();
    hasil = hitungPangkat(bilangan,pangkat);
    System.out.print("Hasilnya adalah = " + hasil);
    System.out.println();
    System.out.printf(format:"Perkalian (%d pangkat %d) = " ,bilangan,pangkat);

    for (int i =0; i<pangkat; i++){
        |   System.out.print(bilangan + " x ");
        |
    }
    System.out.println("1 = " + hasil);
}

```

```

Bilangan yang dihitung: 3
Pangkat: 2
Hasilnya adalah = 9
Perkalian (3 pangkat 2) = 3 x 3 x 1 = 9
PS C:\Users\FITRIA RAMADHANI> 

```

3. Sebutkan blok kode program manakah yang merupakan “base case” dan “recursion call”!

Base case :

```

if (y==0){
    |   return (1);
    |
}

```

Recursion call :

```

else {
    |   return(x*hitungPangkat(x, y-1));
    |
}

```